Review of Computer Engineering Research
2021 Vol. 8, No. 2, pp. 41-63.
ISSN(e): 2410-9142
ISSN(p): 2412-4281
DOI: 10.18488/journal.76.2021.82.41.63

REAL-TIME CAR PARKING SYSTEM USING ARDUINO CONTROL

Ahmed Raza Moshin[1+]
Maira Khalid[2]
Muhammad Awais[3]
Kinza Ahmad[4]

[1,2]*Institute of Southern Punjab, Multan, Pakistan.*
[1]*Email: ahmedraza.1st@gmail.com Tel: +923216337335*
[2]*Email: maira.khalid7272@gmail.com Tel: +923063787406*
[3,4]*Air University Punjab, Islamabad, Pakistan.*
[3]*Email: rjawaissaleem@gmail.com Tel: +923457286380*
[4]*Email: kinzaa210@gmail.com Tel: +923150982427*


(+ Corresponding author)

Article History
Received: 30 August 2021
Revised: 27 September 2021
Accepted: 15 October 2021
Published: 1 November 2021

Keywords
Smart parking
Intelligent parking
Sensors
Internet of things
Ultrasonic sensors
Map Navigation.

ABSTRACT

In the ongoing examination of metropolitan areas, the increment in population produces high vehicle density on roads. Consequently, this prompts irritating issue for the drivers to leave their vehicles as it is hard to discover a leaving space. This paper introduces web based automatic smart parking system for vehicles. In this paper we have proposed a system which can easily manage parking system through networks of different sensors. This system can easily find a parking space and check whether the user parked the vehicle or not. In addition, this system can create unlimited locations and add slots to these locations. This is multi-user parking system where a single application can work for multiple locations. This parking system will improve the probability of successful parking and minimizes the waiting time of user. Moreover, this parking management system will encourage users to track parking slots and make the parking process a hassle-free experience.

Contribution/Originality: This study contributes to the existing IoT literature that uses devices to provide a better parking system. This study investigated IoT with web-based applications and finds the slots for the user with direction provided on screen using map view. This Study shows the documentation of the real time parking system.

1. INTRODUCTION

In many situations, driver faces difficulty to find a vehicle space or check whether vehicle parked at its assigned slot location or not. Most of the time we have to hire more people to check all those problems. To vanquish all these issues, we have to convert the

system to Smart Parking Base System where parking is managed automatically by devices [1]. At this blooming time, we have a lot of Parking Systems present; they all have some advantages like cost effectiveness and some disadvantages like wastage of time. After probe all those situations, we developed a system which is automatic and cost effective. This smart system can be used for real based parking system. We have introduced smart parking system using sensors and communicating devices with Web API. The work is based on web application, where several sensors are connected to Web Application through API. This Web Application is developed in Laravel language. This technology will help to manage parking system easily. New technologies are efficient than previous parking method. Because it contains all the relevant data of vehicles in our database [2] The following type of data about vehicles enlisted below: -

- Automatic scan and generation of token for parking, having the details of parking slot.

- Check in time of vehicle.
- Current location of vehicle to the admin.
- Checkout time of vehicle.
- All the cards used for parking purpose can be recharged on the spot or at the administration block.

2. RESEARCH OBJECTIVE

At this blooming time, we have a lot of Parking Systems present, they all have some advantages like cost effectiveness and some disadvantages like wastage of time. After probe all those situations, we developed a system which is automatic and cost effective. This smart system can be used for real based parking system. Proposed system covers these aspects:

1. This system easily manages parking system through networks of different sensors.
2. This system can easily find a parking space and check whether the user parked the vehicle or not.
3. This system can create unlimited locations and add slots to these locations.
4. This parking system will improve the probability of successful parking and minimizes the waiting time of user.
5. This parking management system will encourage users to track parking slots and make the parking process a hassle-free experience.

3. LITERATURE REVIEW

This section is consisted of last decade methodologies of car parking system using explaining different sensors and algorithms working. Last decade methodologies include paper from 2011 to 2020 with reference of the discussion paper. This may help the researcher to understand the previous working of researchers in this topic and also provide baseline to our project.

3.1. Real Time Car Parking System Using Image Processing (2011) [3]

In this paper, the designed system captures the image of the car in a real time mode. Car is allowed for the parking on the basis of free slots, firstly the left side filled for the parking and after that car are directed towards the right side of the parking. If slots filled completely than no cars are allowed for parking. They use web camera, PC and LEDs as hardware modules and MATLAB as software model for image processing. Methodology consists of following steps:

- Image acquisition.
- RGB to gray conversion.
- Image enhancement.
- Image matching using edge detection.

In the working of system, car image is captured by eb camera and save for the reference image for the parking and RGB to gray conversion is done to enhance the image, gamma correction [4] is used for this conversion. The interval time for the capturing of image is 2 second. After the successful conversion if reference image is matched with real image more than 90%, car is allowed for parking and this parking system accommodate 20 cars at a time.

3.2. Intelligent Parking Space Detection System Based on Image Processing (2012) [2]

In this paper, the designed system is based on image capturing of rounded brown images that is drawn first time manually for the reference of image detection. This technique gives more efficient results of image comparison as compared to use of moving object. This system is using image processing method instead of sensors that may reduce some complexities like wiring and sensor costs. This project is consisted of five modules:

- System Initialization.
- Image Acquisition.

- Image Segmentation.
- Image Detection.
- Image Enhancement.

In the first step of this system, system automatically identifies the parking locations in the parking lot image. The second step system stores the captured image using camera to process this image in the MATLAB. In the third step image is separated from the pixels to improve the contrast by suing threshold technique. In the fourth step image is enhanced using morphological functions to extract extra pixels from image to detect the boundaries of object. In the last step system detects the rounded brown image of each parking lot.

3.3. Real-Time Monitoring System for Parking Space Management Services (2013) [5]

In this paper, the designed system has a GUI that helps the administrator to monitor the status of car parking system in a real time, from the screen interface. This system is based on queue theory and GUI of this system provides other event of simulation like car arrival, departure and other services. The M/M/2 queue model is used to represent the service time and distribution arrival rate for the cars on the entrance gate. The G/G/2/64 model is used on exit gate for the arrival and departure pattern of users, these patterns occur randomly and independently on gate. Through a mathematical model using in this system the probability of having vehicles is calculated that may be zero or for any specific number. To analysis the queue system performance the formulate values can be determined using parameters include expected (average queue length, average number in the system, average total time and average waiting time). This system is consisted of two subsystems:

- Entrance Barrier Gate
- Exit Gate

The entrance system work like outer network for servicing time of cars and ticket machines. There are 2 barrier gates acts as servers G1 and G2. The queue of cars serves FIFO technique on the gates. The exit gate system uses G/G/2/64, consists of 2 servers on each exit gate E1, E2. The number 64 depicts that the parking lot have the capacity of 64 cars for parking in the zone. Queue system ensures that parking system can never be exceed from parking lot range, it first fills first level and then directs the car to the second level. The proposed system of this paper use Dijkstra Algorithm for the implementation of the model. Through this algorithm all possibilities are calculated for parking.

3.4. Integration of RFID and WSN Technologies in a Smart Parking System (2014) [6]

In this paper, the designed system is based on the combination of different IoT technologies including RFID (Radio Frequency Identification), WSN (Wireless Sensor Network), UHF (Ultra High Frequency) and a mobile. The proposed system is consisted of heterogenous network of hybrid RFID, IEEE802.15.4-based WSN and UHF devices that can easily be implemented on any outdoor parking system. System uses RESTful java software application that monitors parking lot with central implemented database management system. The proposed uses mobile application to direct the drivers to the nearest vacant parking location. And also notify the occupancy of parking. Moreover, system uses NFC based e-wallet to provide the facility to users for paying parking charges. GCM (Google Cloud Messaging) is installed on central servers to manage the alert events like expiration of purchased time and improper use of parking). This information is displayed to cop by mobile applications to control this ad hoc scenario. This system is consisted of:

- WSNs.
- SG (Smart Gateway).
- CS (Central Server).
- Parking App for drivers.
- Policemen App for traffic cops.

The main components of proposed Zigbee network are R (Router) and C (Coordinator) nodes. R nodes have the information for routing capabilities and C nodes collect information and send to central server. WSN with different R and RR (Router Reader) used in this system. R nodes are placed on parking lot with light sensors to monitor the state, whereas RR nodes are placed on reserved parking poles. The retrieved information from nodes send to C nodes in a multi-ho manner, which is then transmit to SG. All the information is analyzed and transmit to CS along with parking position. The NFC based payment is provide to user for parking the vehicle, provided by SG. The main work of RR nodes is used to check the authorized car parking labeled by UHF-RFID tags. The database handling is done by CS that is used to manage all payment of users and parking spaces. Expiration and reserved details of parking is displayed on drivers and police Aps with the help of Google Cloud Message services.

3.5. Park Here! A Smart Parking System based on Smartphones' Embedded Sensors and Short Range Communication Technologies (2015) [7]

In this paper, the designed approach uses smart phone to detect automatic parking without using on spot sensors. Sensors embedded in smart phones like gyroscope, accelerometer and Bluetooth connectivity is used in the proposed with algorithm for detection. Combination of internet connections, Wi-Fi links and D2D (Device-to-Device) connections allows the strategy to regulate the information about parking. Main components of the system are:

- Parking event detection
- Parking data repository
- Parking data dissemination
- Parking data fusion

Parking event detection is the triggering event of the user side that triggers parking actions of current user. This action can be done in the background without involving user interaction. Parking data repository checks the statistics and the recent actions triggered by the current vehicle or the vehicle in the range. Repository is updated for the CD (City Database) vehicle to perform vehicle periodic access. Repository is updated through data received by mobile over Wi-Fi connections.

Parking data dissemination updates the repository by vehicle information and notify the users using internet connection, this feature is a bottleneck of system because the load on the internet led down the system. Local dissemination is performed by D2D of mobile using Park Her! Application to detect the parking for a vehicle. Parking data fusion perform merging to the data stored in local repository. In the rest of the paper algorithms and the results are discussed on the basis of successful experimentation.

i Parker—A New Smart Car-Parking System Based on Dynamic Resource Allocation and Pricing (2016) [8]

In this paper, the proposed is based on MILP model that provide the parking dynamically and statically for the users, and providing variety of reservation options. The proposed system of this paper combines different reservation techniques for parking includes Share-time, Real-time and Dynamic. This system also proposes pricing policies for dynamic and static reservations and provide maximize profit for parking management. The proposed system is consisted of two types of reservations:

- RTR (Real Time Reservation)
- STR (Share Time Reservation)

With this combination driver can reserve parking in advance or on a same time. RTR uses dynamic resource allocation and provide drivers the best parking until they reached at parking spot. While, STR uses static allocations of time frames and allocate the parking to the drivers. Dynamic price engine is used to update the prices for parking on the basis of resource allocation. iParker framework includes:

1. Authority
2. Parking Manager

3. Pricing Engine
4. SAC (Smart Allocation Center)
   i. Static Allocation
   ii. Dynamic Allocation
5. Data Centre
6. Sensors
7. Virtual Message Signs
8. Central Request Center
9. Parkers

3.6. Advanced CAR Parking System using Arduino (2017) [9]

In this paper, the proposed architecture design is based on Arduino control. The proposed system is allowed to park a car only to authorized person using a card containing information about the vehicle number and other details. If the user is authorized, he can park his car only if the parking slot is free, if parking is busy, he is unable to park his car even he is authorized. If user parked his car in parking mobile notification is generated. The parking system is not allowed to park a car by unauthorized user. This parking system is a multifloored parking system and display the free parking slot on each floor.

The proposed system uses IR sensors on parking slot to sense any obstacle and vehicles. Sensor's information is displayed on the Entry and Exit gate, if use's RFID is matched with the punched card, it allows user for entry and exit. Database is maintaining to check the user's entry and exit. There are two gates used in the parking:

- Entry
- Exit

On the entry gate, user punches his RFID card on RFID Reader, Reader will read the tag information and send it to the Arduino for authorization verification. After receiving the valid authorization, signals will send to electric motor for opening the gate. There is a counter variable which is incremented and decremented according to car entry and exit on parking.

On the exit gate, user again punches his RFID card on RFID Reader, IR sense the vehicle and decrement the value by 1 if he is authorized user and open the gate by sending signal to electric motor. All information of increment and decrement is display on LED screens in the parking.

I-SPARK: IoT based Smart Parking System (2018) [10]

In this paper, the proposed system is consisted of great combination of IoT technology and webpage or mobile device to display the parking slots. The proposed system uses RFID for automatic billing and security purposes as well. Features of system involves:

- IR Sensor
- LM35 Sensor (Temperature)
- LDR Sensor (Light)
- Arduino Mega+ Ethernet Shield
- Cloud
- HTML Page/ Mobile
- Commands

These features are the combination of both software and hardware. Further, this system is divided into two parts:

- Monitor the empty slots
- Monitor the light and temperature for concerned person.

Both cases include cloud due to its availability in the protocol of MQTT that accomplish the core element tasks of real time system for servers, client, and topics [11]. Monitoring of temperature and light requires LM35 and

LDR sensors and their data after processing will send to the authorized person and indicate about darkness and temperature change. Monitoring of empty slots is accomplished by IR/Ultrasonic sensors. IR sensors are being used for indoor parking and ultrasonic sensors are used for outdoor parking. Webpage indicates about the free slots to user, filled slots are highlighted through red spot on the screen.

3.7. A Real-Time Cloud-Based Intelligent Car Parking System for Smart Cities (2019) [12]

In this paper, the proposed system is based on real-time cloud-based IoT technologies that operates without image processing. To enhance the security in the system different MAC addresses are used for routers and wireless sensors placed on different parking places along Data Encryption Standard (DES). Proposed system architecture includes:

- Wireless Sensors
- Wireless Router
- Integrated Cloud Platform
- Digital Parking Signs
- Web Apps and Analytics
- Mobile Apps

The status of the proposed real-time status is detected by WSNs and sends to the wireless router. Wireless routers are cost effective for the large-scale parking. In this parking system user can login in to the system and select the slot for parking. The slot status change to the "pending" after user's selection, if user not pay the fess of parking within two minutes, it will again change the status of "available". The status of parking is updating through WSN nodes and the parking slots selection can be accomplished by different set of notification steps done by user.

3.8. A Real-Time Automatic Plate Recognition System Based on Optical Character Recognition and Wireless Sensor Networks for ITS (2020) [13]

In this paper, proposed is based on Optical Character Recognition (OCR) of number plates of vehicle. This system is considered more efficient due to excessive use of traffic cameras, because almost all cameras ae connected through internet and using vision techniques slots for the parking can be detected easily. This system captures the image of number plates of vehicle and verify its licensed and also detect parking slot of the vehicle. The verification process of number plate is accomplished in real-time mode and stored in database for the record of traffic department and agencies. The proposed system architecture includes:

- Plate Localization (Object Detection)
- Plate Extraction
- License Plate Segmentation
- Character Recognition

Plate Extraction and License Plate Segmentation are the preprocessing steps and Character Recognition implement by Pytesseract (Python-tesseract). The algorithm of proposed system is start with plate localization that extract license plate from the image captured by camera. It applies bilateral filter [14] to reduce the noise of captured image for accurate detection of number plate as it effects by the environment as well. Moreover, system calculates the data on the basis of defined parameters of accuracy.

4. PROPOSED INFRASTRUCTURE

The foremost contemplate of parking Management System is to diminish the manual work and waiting time of the customer. Having the purpose of, various methodologies are used in parking system [15]. The parking system that we flourish is a concoction of various technologies.

List of these technologies are expounding below:

1. Web Application
2. Devices

1. Web application

The whole system is working around a web application. There is admin panel that uses API to use web technologies.

a. Admin Panel

Admin panel is a control panel where admin can create location and slots related configuration. Admin can manage new locations and slots according to the requirements, shows in use case diagram of admin (Figure 1).

Functionalities:

- Create Locations:

Admin can create multiple location from the admin panel.

- Create Vehicle Types:

Admin can create vehicle types that are allowed for parking place.

- Create Slots:

Admin can create slots.

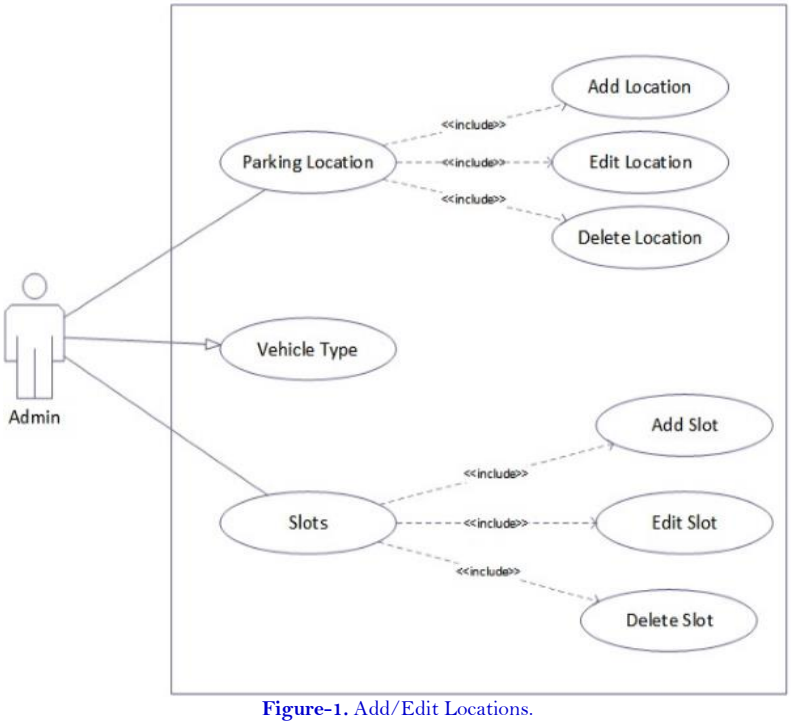

Figure-1. Add/Edit Locations.

- Slot Descriptions:

Admin can check parked car on slot.

- Configuration:

Admin can set configuration related to location on map for each slot with a unique labeled name.

b. API

We developed Web Based APIs for communication between other resources and Web Application. **API URL** is used for updating the status of slot, as the slot is free or occupied.

c. Checkpoint Entry Page

Entry page is loaded on the screen when vehicle arrives at entry point.

Figure-2. Main Entry Page.

Driver will see the page (Figure 2) to log his/her vehicle, click on button, a URL will open that send request to camera to take picture and then with that picture this request work internally and find an empty slot. If empty slot will find. It saves image against that slot and return the address of that slot to the user screen Figure 3.

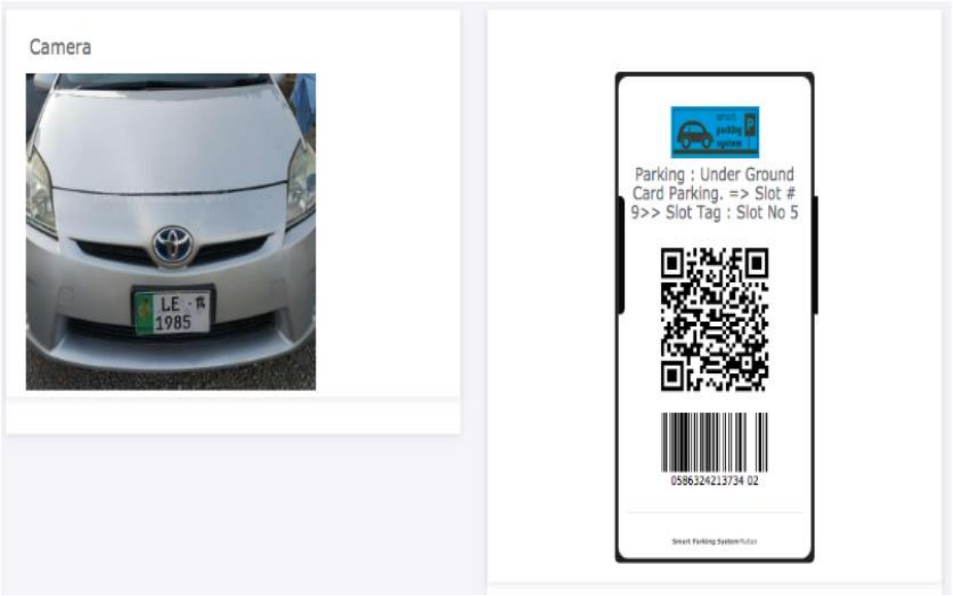

Figure-3. User Screen.

Then the address of the slot will return by system, that will be printed on a slip with QR Code and a reference generated the Barcode [16] That QR Code is linked to the slot location Figure 4.

Figure-4. QR Code of Location.

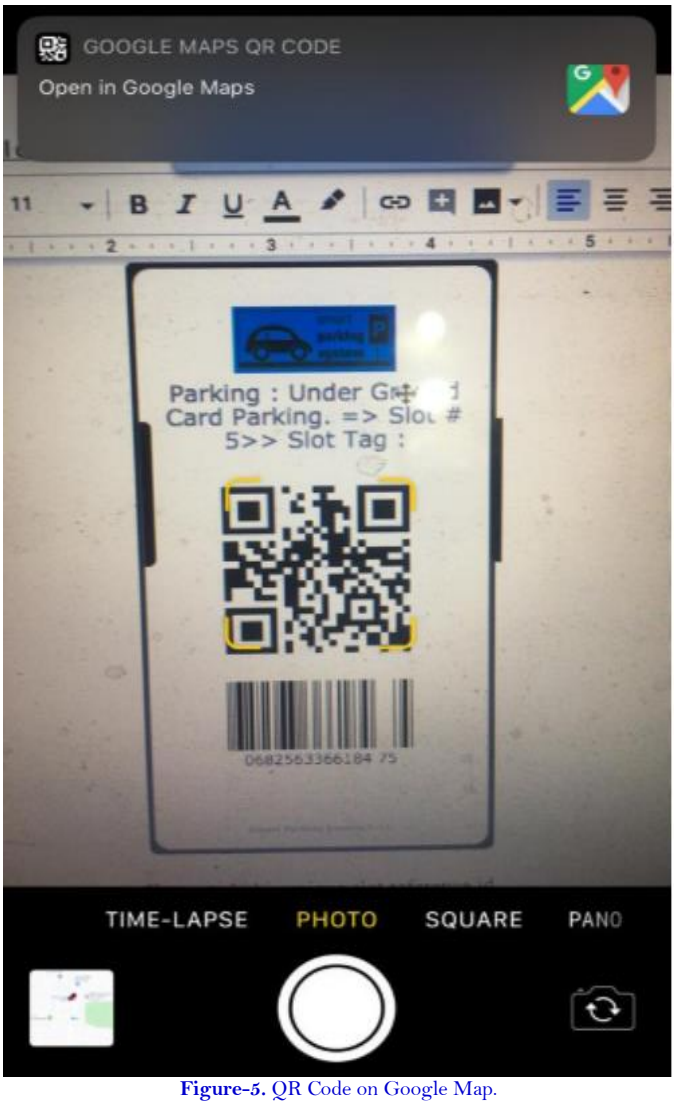

Figure-5. QR Code on Google Map.

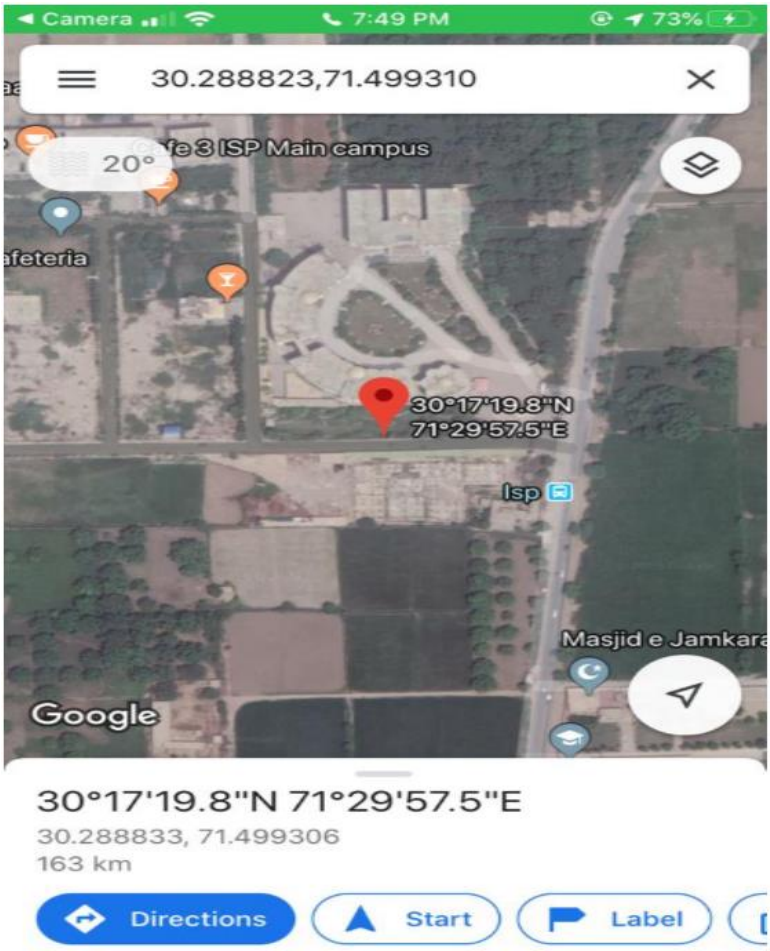

Figure-6. Google Map View of Slot.

When user scan QR code (Figure 5), it will open Google Map application in the mobile phone (Figure 6) and direct the location of parking slot [17].

2. Devices

The devices included in the proposed methodology are given below:

a. Ultrasonic sensor

An Ultrasonic sensor (Figure 7) is a device that is used to measure the object distance by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object [18].

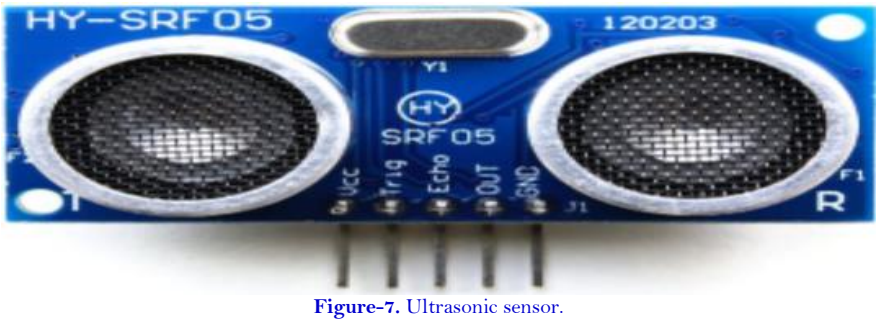

Figure-7. Ultrasonic sensor.

b. Arduino Uno

The Arduino Uno (Figure 8) is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button [19].

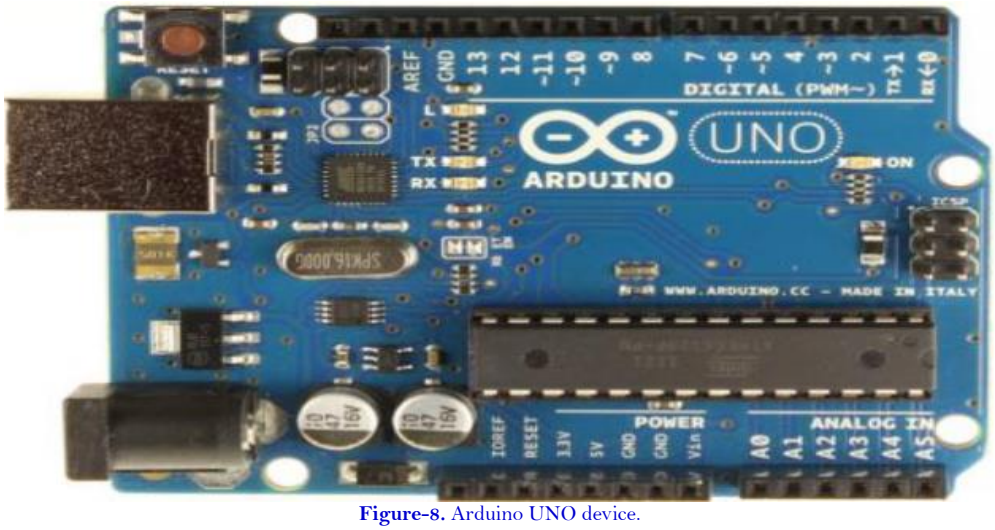

Figure-8. Arduino UNO device.

c. Node MCU

NodeMCU (Figure 9) is an open source IoT platform. It includes firmware which runs on the ESP8266 shown in (Figure 11) Wi-Fi SoC from Expressive System, and hardware which is based on the ESP-12 module [20]

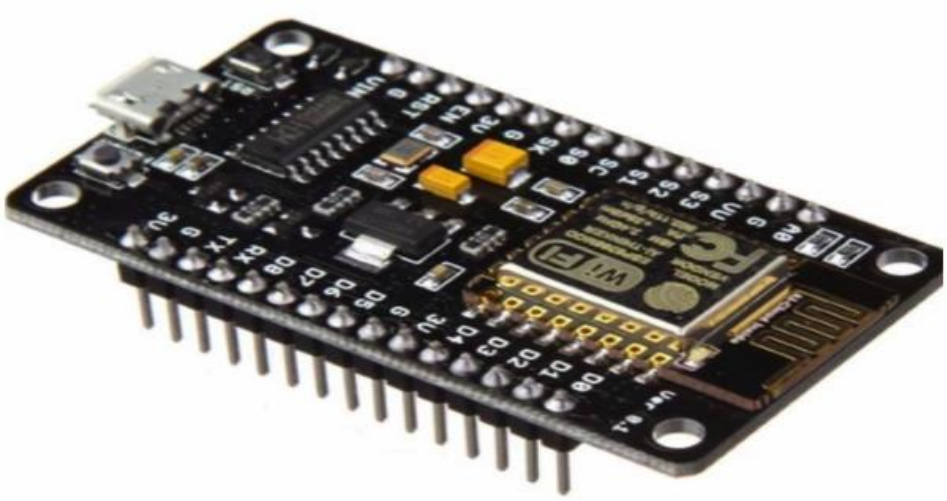

Figure-9. NodeMCU device.

d. Breadboard

Breadboards are designed to work with through-hole electronic components (Figure 10). These components have long metal leads that are designed to be inserted through holes in a printed circuit board (PCB) that are plated with a thin copper coating, which allows the components' leads to be soldered to the board [21].

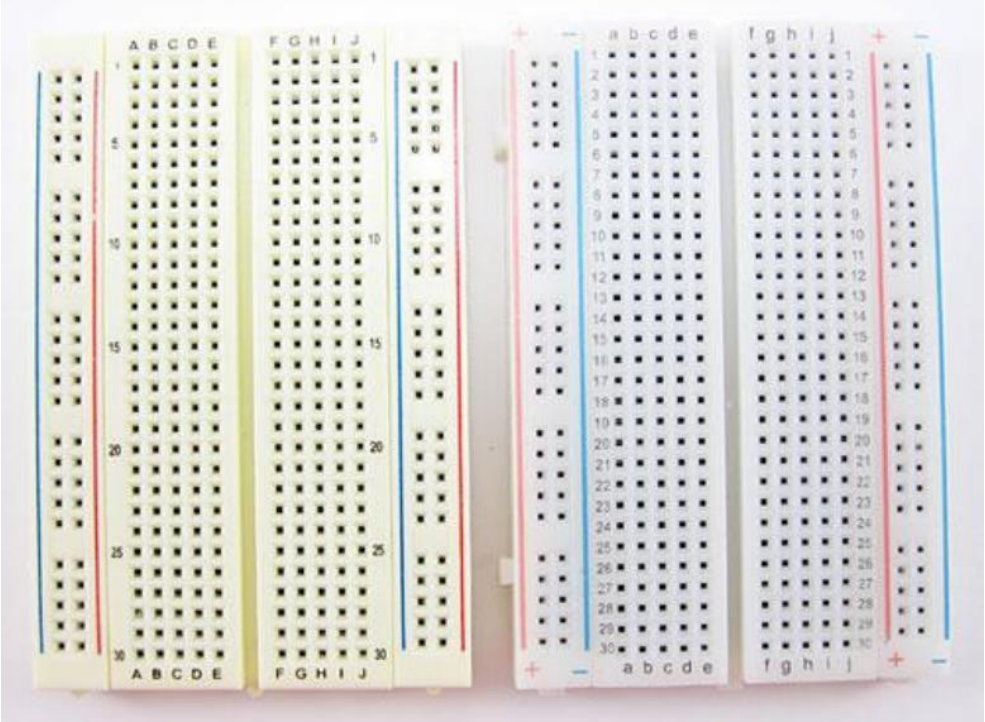

Figure-10. Breadboard Circuit.

e. Communication Channel

This is a low-cost Wi-Fi microchip with full TCP/IP stack [22],

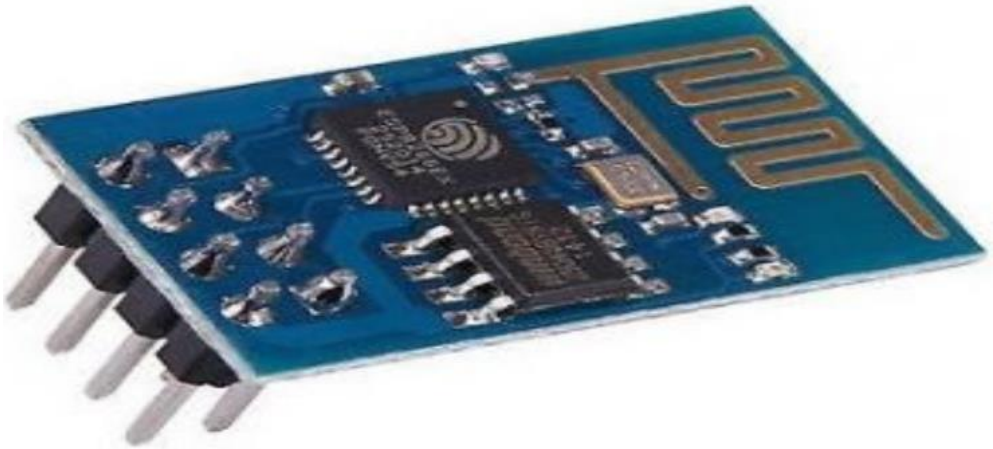

Figure-11. Microchip ESP8266.

f. Camera

Two-dimensional imaging scanners are a newer type of barcode reader. They use a camera and image processing techniques to decode the barcode. Camera is also used for security purposes.

5. WORKFLOW

In this work, there is a web application that updates the availability of different parking slots in the respective areas. To implement this, system uses Laravel server using database SQL server [23]. There is a barcode scanner at the entry point that is connected to system, user scans his/her card Figure 12 and camera capture his/her vehicle number plate then send to system for processing and a slot is assigned against that vehicle and user. On exit point user again scans his/her card and those occupied slots are freed for that vehicle. This system will encourage user to

use parking system and make parking process a Hassle-free experience Figure 14. In future, this application flexibility can be improved by sending the registered id to the user as a text message or email and alerts about the slot information. Also, in future we will connect sensors to detect vehicle on slots.

Figure-12. Scanning User Card.

In this parking system, we have the status of slots i.e., is booked or free for web application. When vehicle reached at the slot, the system will update the status of slot as booked and when vehicle will exit from the slot the status will updated as FREE. This process is done by calling API connected to Internet by NodeMCU. System has different details with respect to actors and actions discussed below.

1. Admin Dashboard

With Admin dashboard we can do following actions

- Dashboard: Parking Detail about Locations count, Slots count, Charged Amount, Parking User count and Security User.
- Admin can add/edit vehicle type Vehicle Type
- Admin can add/edit user categories
- Admin can add/edit/block Parking location. Admin can set map for parking location and can add slots parking location. Admin can set latitude and longitude for parking location
- Admin can add/edit/block security users that are working for this parking management on enter/exit check point.
- Admin can add/edit/block parking users. Also, can add vehicles to that user.
- Admin can check Parking Detail like vehicle number, username amount deducted date and charged amount.

Features of proposed system is given in the below table (Table 1) that explains all occurrences of features:

Table-1. Features of proposed system.

Features	Proposed System Occurrence
QR code Scanning system	✓
Unlimited Vehicle Types	✓
Free and Paid Parking user type	✓
Auto Read Vehicle number plate	✓
Dynamically creating Locations and Slots	✓
Parking Card Scanning	✓

2. Security User Panel

i. Security user panel have authority for parking user card scanning on entry point.
ii. After scanning card with Barcode reader camera capture the image of vehicle.
iii. If image is not readable than popup to ask user enter plate number annually.
iv. A popup show detail for slots of parking location assigned to user. And also, a QR code for user to get map location of parking location on mobile.
v. Exit point is for scanning card on exiting.

3. Check Point In

i. Capturing Image after scanning card.
ii. If number plate is unable to read then this popup will appear for entering manually parking number.
iii. If the vehicle already parked there then this popup will appear.
iv. If vehicle number plate number is correct and card scanned was correct then successfully a lot will be assigned. In popup it will show location detail and location map link in QR code. Figure 13

Figure-13. QR Location Screen.

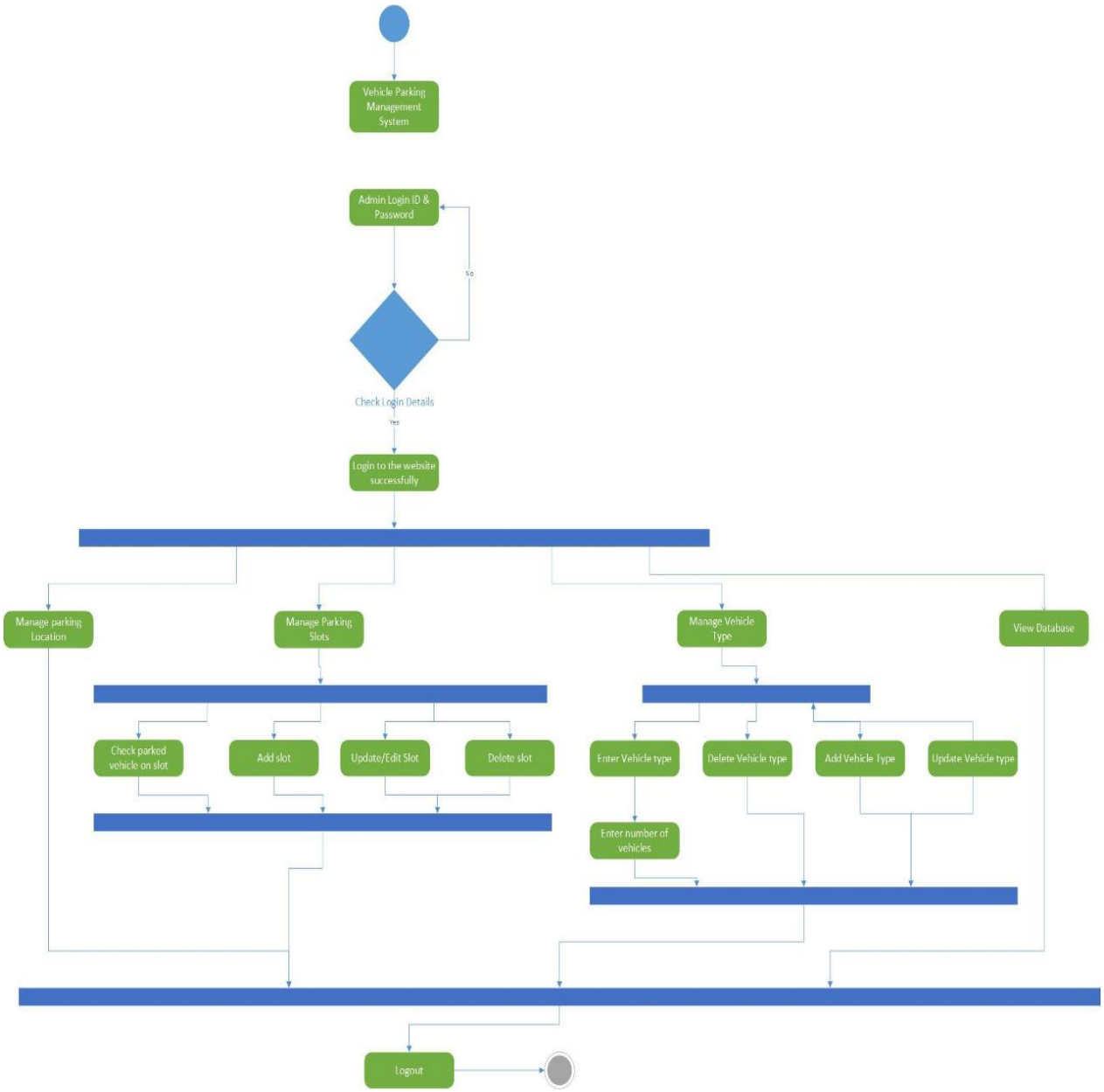

Figure-14. Workflow of System.

4. *Check Point Out*

i. Slot will free after successful checkout.

5. *Parking User*

i. Check account balance.

ii. Parking user can download his parking card.

iii. Can check all vehicles number plates added for his account.

6. *API URL*

Sensor-slot-api/ (slot_id from database)/ (status 0 or 1)

When vehicle leave the slot, this URL will call

http://127.0.0.1:8000/sensor-slot-api/1/0

When vehicles enter into the slot this URL will call

http://127.0.0.1:8000/sensor-slot-api/1/1

Sensor will detect the vehicle at the slot, it passes the signals to NodeMCU for vehicle detection and call API to change the status of the slot in Web Application. When vehicle leave the slot, sensor will again detect the empty space that there is no vehicle, it passes the signal to NodeMCU and it will again call API URL [24] to update status of slot in Web Application. The flow of devices is shown on the given figure. (Figure 15)

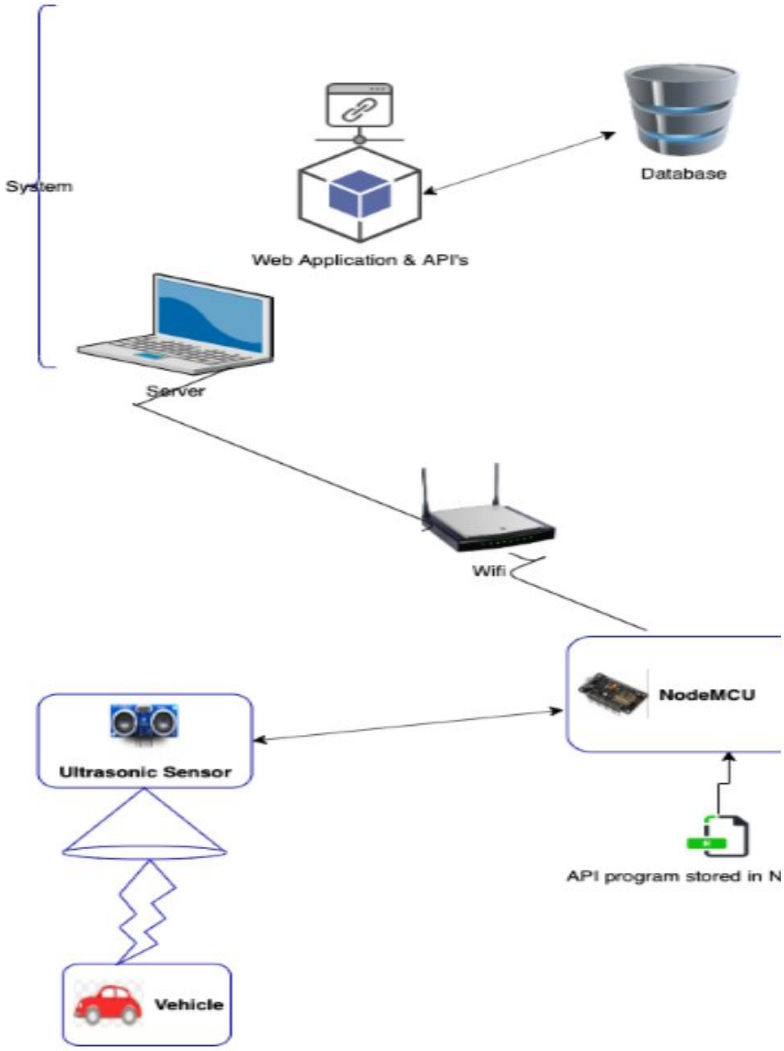

Figure-15. Flow of Device.

7. Database

In this system, we have following Database tables shown in (Figure 16)

1. Admin Table
2. Institute Table
3. Locations Table
4. Management Table
5. Slot Table
6. Slot Vehicle Table
7. Vehicle Table
8. Vehicle Type Table

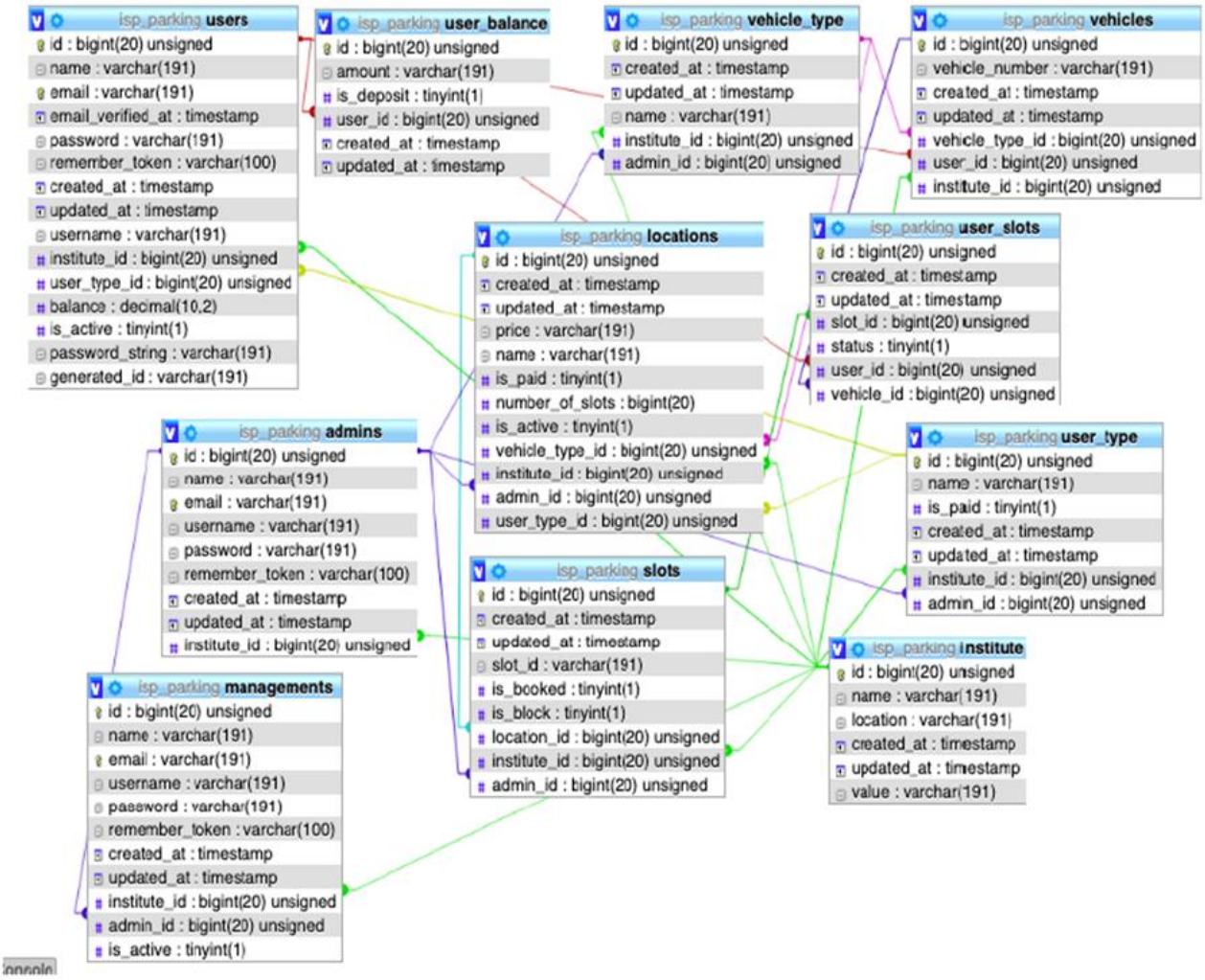

Figure-16. Database Schema.

8. Test Cases Test Case Title: Login forms

Test case1: Login Form

Admin Login, Parking User Login, Security User login

Priority (H, L): High

Test objective: Verifying Login

Test descriptions: "User enter the required field, press the submit button", client program contact to the server, server contact with database, database check the result and send the results to the user.

Requirements verified: Yes

Test environment: MySQL must be in running state. Database should contain appropriate table and link must be established between the server and client program.

Test setup/ pre condition: MySQL and Laravel environment must be in running state. The entire mandatory field must be entered.

Location

Expected result

The user will login to access Parking System.

Display main menu.

Pass: Yes Condition: Pass Fail: No

Problem /issues: NIL

Status: Successfully Executed

Test Case Title: Security User/Parking User Registration forms.
Test case2: Security User/Parking User Registration
Security User/Parking User Registration forms
Priority (H, L): High
Test objective: Verifying Registering

Test descriptions: "User enter the required field, like email, password, phone no, username etc. press the submit button", client program contact to the server, server contact with database, database check the result and send the results to the user.

Requirements verified: Yes

Test environment: MySQL must be in running state. Database should contain appropriate table and link must be established between the server and client program.

Test setup/ pre condition: MySQL and Laravel environment must be in running state. The entire mandatory field must be entered.

Location
Expected result
The user will access to display page for users
Display the content page
Pass: Yes Condition: Pass Fail: No
Problem /issues: NIL
Status: Successfully Executed

Test case title:

Test case 3: Tested checkpoint in/ checkpoint out with image reading captured by camera and card scanning
Priority (H, L): High
Test objective: Verifying checkpoint in/ out

Test descriptions: At checkpoint User will scan card to Barcode reader, request will be sent to camera to capture the image. Image will send to API Cloud. After processing image, a number plate number read from image will be sent as response. System will verify that vehicle number and issue slot again her card. At Checkpoint out user will scan card and his/her assigned IOT will be freed and amount will be deducted from his card according to fee setup by admin.

Requirements verified: Yes

Test environment: MySQL must be in running state. Database should contain appropriate table and link must be established between the server and client program.

Test setup/ pre condition: MySQL and Laravel environment must be in running state. The entire mandatory field must be entered.

Location
Expected result
The user will access to display page for users
Display the content page
Pass: Yes Condition: Pass Fail: No
Problem /issues: NIL
Status: Successfully Executed

6. RESULTS AND EVALUATION

In the workflow it is already mention that barcode scanner and a camera is connected with a computer system/ laptop on Checkpoint IN there is input field where generated code will be appended after card scanning.

Then a request will be sent to server for verifying card. If card is verified then the system will prompt the user to enter number plate only if the system cannot read number plate from captured image. If the system successfully read number from plate, then system will allocate a slot against that user and will show location detail along with map URL in code.

Prompt will take input as number plate shown in Figure 17.

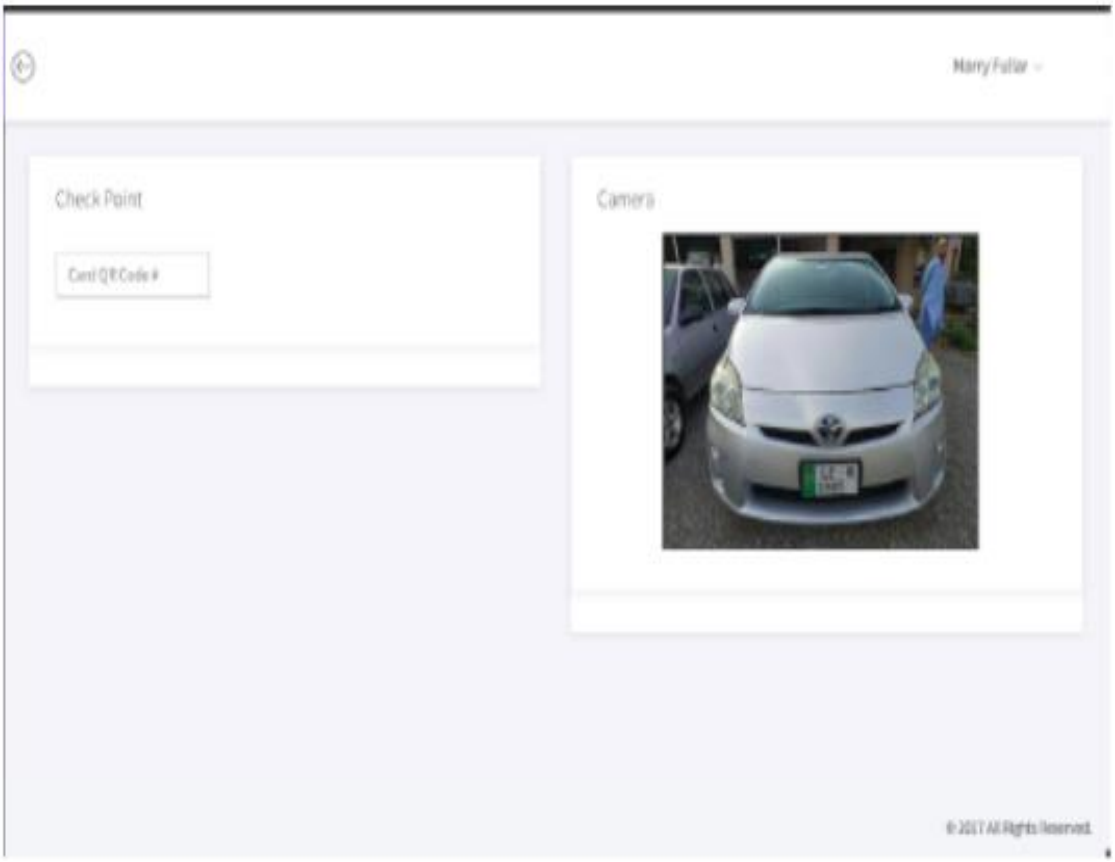

Figure-17. Prompt Screen.

If the number plate reading is failed it will show error mentioned in Figure 18.

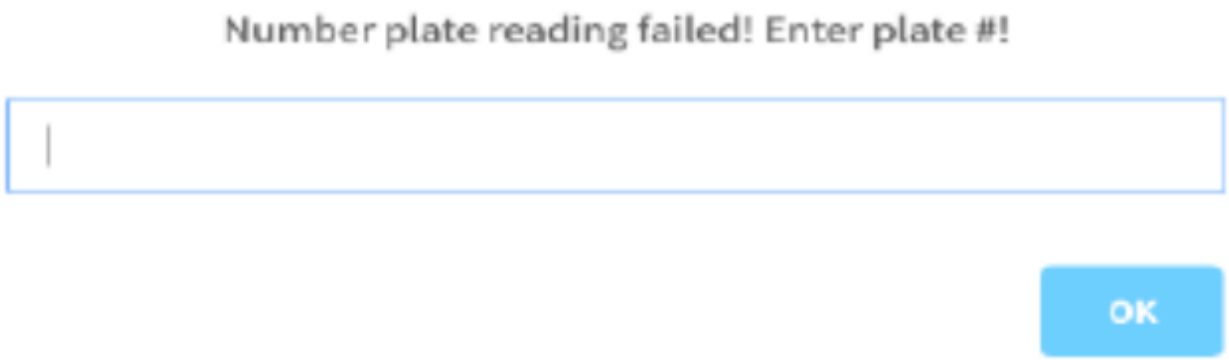

Figure-18. Number Plate Popup.

If users want to go to that location if he/she cannot find then use have to scan this QR in mobile and google map will be opened for user to pointing to location where slots are assigned. After allocating slot given popup will appear with slots detail and QR code for map shown in Figure 19.

Figure-19. Parking Location Popup.

Map will be opened after scanning the barcode and mobile device or scanning device shows the directions shows in Figure 20.

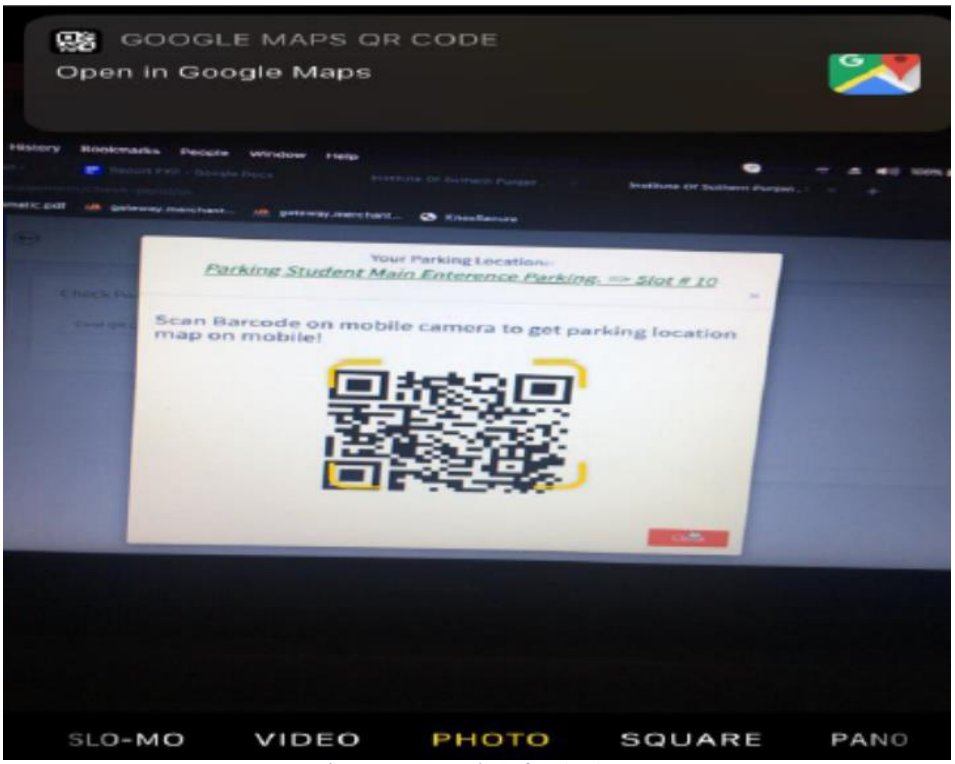

Figure-20. Map View after QR Scan.

Finally, the map location with the direction's details is as follows shown in Figure 21.

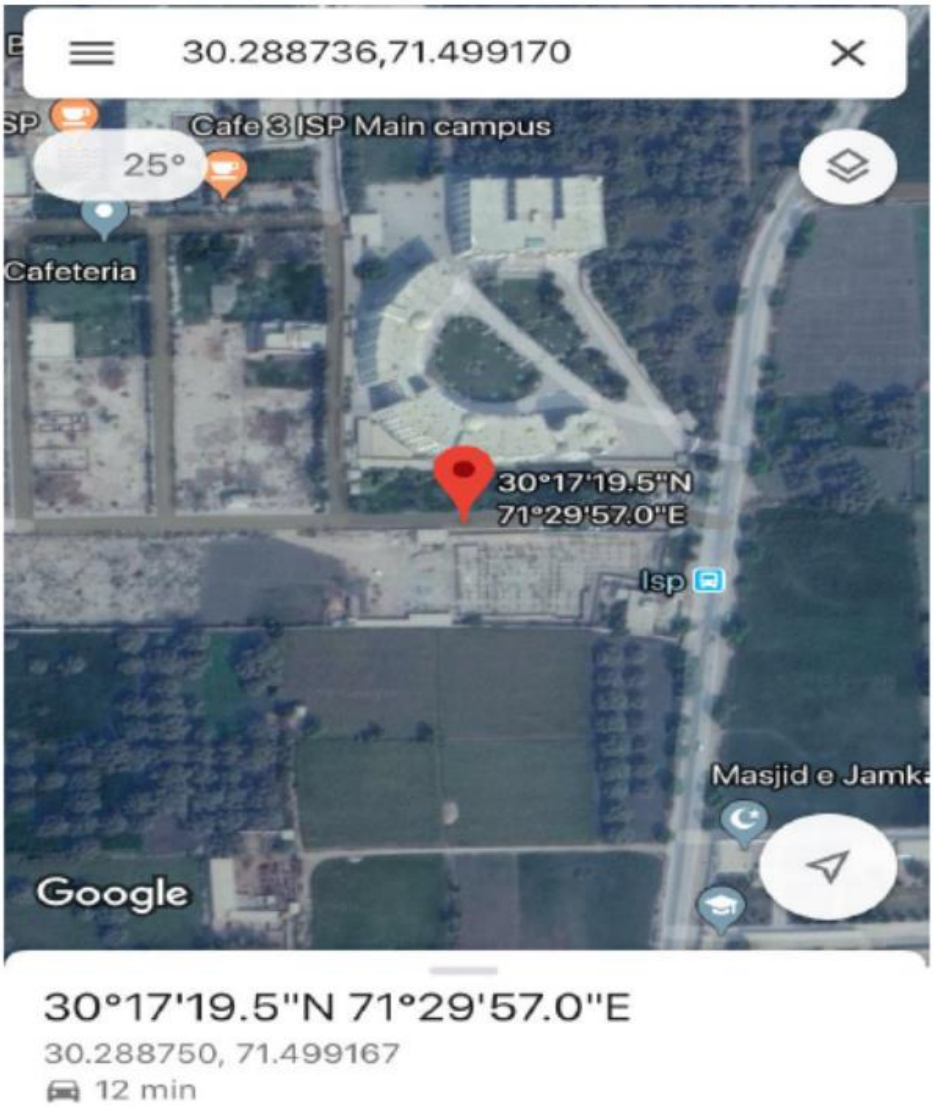

Figure-21. Final Map View.

7. CONCLUSION

This paper is the implementation of a smart parking management system. In this work, we have developed a web application to check the availability of different parking slots in the respective areas. To implement this, we have used Laravel server using database SQL server [23]. There is a scanner at the entry point that is connected to system; user scans his/her card and camera capture his/her vehicle number plate then send to system for processing and a slot is assigned against that vehicle and user. On exit point user again scans his card and that occupied slot is freed for that vehicle. This system will encourage user to use our parking system and make parking process a Hassle-free experience. In future, this application flexibility can be improved by sending the registered id to the user as a text message or email and alerts about the slot information. Also, in future we will connect sensors to detect vehicle on slots.

8. FUTURE WORK

In future we will enhance the implementation using ultrasonic sensor. This sensor will detect the presence of vehicles at each slot. If vehicle is detected then through API, it will update status to database. With ultrasonic Sensors and Arduino chips we will implement the automatic detection and requesting to camera to take picture of

vehicle after the arrival of vehicle. Moreover, we will work on android application for online booking of parking slot. At the entrance of parking system, we will implement biometric login for the user e.g., face login. User can login to their account by face detection with login credentials [25].

Funding: This study received no specific financial support.
Competing Interests: The authors declare that they have no competing interests.
Acknowledgement: All authors contributed equally to the conception and design of the study.

REFRENCES

[1] M. Anila, "implementation of smart parking management system using apache tomcat server," *International Journal of Advanced Research in Computer Science*, vol. 8, pp. 549-553, 2017.Available at: https://doi.org/10.26483/ijarcs.v8i9.5182.

[2] R. Yusnita, F. Norbaya, and N. Basharuddin, "Intelligent parking space detection system based on image processing," *International Journal of Innovation, Management and Technology*, vol. 3, pp. 232-235, 2012.

[3] S. Banerjee, P. Choudekar, and M. Muju, "Real time car parking system using image processing," presented at the In 2011 3rd International Conference on Electronics Computer Technology. IEEE, 2011.

[4] H. Farid, "Blind inverse gamma correction," *IEEE Transactions on Image Processing*, vol. 10, pp. 1428-1433, 2001.Available at: https://doi.org/10.1109/83.951529.

[5] T. P. Hong, "Real-time monitoring system for parking space management services," presented at the In 2013 IEEE Conference on Systems, Process & Control (ICSPC). IEEE, 2013.

[6] L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and R. Vergallo, "Integration of RFID and WSN technologies in a smart parking system," presented at the In 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE., 2014.

[7] R. Salpietro, L. Bedogni, M. Di Felice, and L. Bononi, "Park Here! a smart parking system based on smartphones' embedded sensors and short range Communication Technologies," presented at the In 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE, 2015.

[8] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, "iParker—A new smart car-parking system based on dynamic resource allocation and pricing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2637-2647, 2016.Available at: https://doi.org/10.1109/tits.2016.2531636.

[9] H. Chaudhary, P. Bansal, and B. Valarmathi, "Advanced CAR parking system using Arduino," presented at the In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2017.

[10] P. Chippalkatti, G. Kadam, and V. Ichake, "I-SPARK: IoT based smart parking system," presented at the In 2018 International Conference On Advances in Communication and Computing Technology (ICACCT). IEEE, 2018.

[11] M. Collina, M. Bartolucci, A. Vanelli-Coralli, and G. E. Corazza, "Internet of Things application layer protocol analysis over error and delay prone links," presented at the In 2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC). IEEE, 2014.

[12] F. Mohammadi, G.-A. Nazri, and M. Saif, "A real-time cloud-based intelligent car parking system for smart cities," presented at the In 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP). IEEE, 2019.

[13] N. d. V. Dalarmelina, M. A. Teixeira, and R. I. Meneguette, "A real-time automatic plate recognition system based on optical character recognition and wireless sensor networks for ITS," *Sensors*, vol. 20, p. 55, 2020.Available at: https://doi.org/10.3390/s20010055.

[14] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," presented at the In Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271). IEEE., 1998.

[15] W. J. Park, B. S. Kim, D. E. Seo, D. S. Kim, and K. H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," presented at the In 2008 IEEE intelligent vehicles symposium. IEEE, 2008.

[16] M. Kikuchi, M. Fujiyoshi, and H. Kiya, "A new color QR code forward compatible with the standard QR code decoder," in *2013 International Symposium on Intelligent Signal Processing and Communication Systems (pp. 26-31). IEEE*, 2013.

[17] A. K. Akanbi and O. Y. Agunbiade, "Integration of a city GIS data with Google Map API and Google Earth API for a web-based 3D geospatial application," *arXiv preprint arXiv:1312.0130*, 2013.

[18] J. Liu, J. Han, H. Lv, and B. Li, "An ultrasonic sensor system based on a two-dimensional state method for highway vehicle violation detection applications," *Sensors*, vol. 15, pp. 9000-9021, 2015.

[19] Y. A. Badamasi, "The working principle of an Arduino," presented at the In 2014 11th International Conference on Electronics, Computer and Computation (ICECCO) . IEEE, 2014.

[20] L. Shkurti, X. Bajrami, E. Canhasi, B. Limani, S. Krrabaj, and A. Hulaj, "Development of ambient environmental monitoring system through wireless sensor network (WSN) using NodeMCU and "WSN monitoring"," presented at the In 2017 6th Mediterranean Conference on Embedded Computing (MECO). IEEE., 2017.

[21] M. Tawfik, E. Sancristobal, S. Martin, R. Gil, G. Diaz, A. Colmenar, J. Peire, M. Castro, K. Nilsson, and J. Zackrisson, "Virtual instrument systems in reality (VISIR) for remote wiring and measurement of electronic circuits on breadboard," *IEEE Transactions on Learning Technologies*, vol. 6, pp. 60-72, 2012.Available at: https://doi.org/10.1109/tlt.2012.20.

[22] A. M. Abd Jalil, R. Mohamad, N. M. Anas, M. Kassim, and S. I. Suliman, "Implementation of vehicle ventilation system using NodeMCU ESP8266 for remote monitoring," *Bulletin of Electrical Engineering and Informatics*, vol. 10, pp. 327-336, 2021.Available at: https://doi.org/10.11591/eei.v10i1.2669.

[23] M. Anif, A. Dentha, and H. W. S. Sindung, "Designing internship monitoring system web based with Laravel framework," presented at the 2017 IEEE International Conference on Communication, Networks and Satellite (Comnetsat) (pp. 112-117). IEEE, 2017.

[24] H. G. C. Ferreira, E. D. Canedo, and R. T. De Sousa, "IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and arduino," presented at the In 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, 2013.

[25] N. Yıldırım and A. Varol, "Android based mobile application development for web login authentication using fingerprint recognition feature," presented at the In 2015 23nd Signal Processing and Communications Applications Conference (SIU). IEEE, 2015.